High Performance Nap of-the-Earth Unmanned Helicopter Flight

Eric N. Johnson, eric.johnson@ae.gatech.edu, Lockheed Martin Associate Professor of Avionics Integration

John G. Mooney, john.g.mooney@gatech.edu, Graduate Research Assistant

Chester Ong, gtg177i@mail.gatech.edu, Graduate Research Assistant

Georgia Institute of Technology, Atlanta, Georgia

Vineet Sahasrabudhe, vsahasrabudhe@sikorsky.com, Manager, Handling Qualities and Control Laws

and Jonathan Hartman, jonathan.hartman@sikorsky.com, Engineer, Advanced Concepts

Sikorsky Aircraft Corporation, Stratford, Connecticut

This paper describes recent results from a partnership between the Sikorsky Aircraft Corporation and the Georgia Institute of Technology to develop, improve, and flight test a sensor, guidance, navigation, control, and real-time flight path optimization system to support high performance nap-of-the-Earth helicopter flight. The emphasis here is on optimization for a combination of low height above terrain/obstacles and high speeds. Multiple methods for generating the desired flight path were evaluated, including (1) a simple processing of each laser scan; and (2) a potential field based method. Simulation and flight test results have been obtained utilizing an onboard laser scanner to detect terrain and obstacles while flying at low altitude, and have successfully demonstrated obstacle avoidance at speeds up to 40 ft/s while maintaining a miss distance of 50 ft horizontally and vertically. These results indicate that the technical approach is sound, paving the way for testing of even lower altitudes, higher speeds, and more aggressive maneuvering in future work.

Introduction

Unmanned aerial vehicles (UAVs) and optionally piloted aircraft are expected to play an increasingly important role in both civil and military applications. Military applications include, among others, intelligence, surveillance and reconnaissance (ISR), cargo transport, and armed attack mission profiles. A specific challenge for military unmanned helicopters is reducing vulnerability of the aircraft during operations. Vulnerability reduction through Nap-of-Earth flight (low altitude, high speed) is a currently accepted tactic for manned military helicopters and an appealing choice for unmanned variants. For a manned aircraft, NOE flight is characterized by the need for a skilled human operator utilizing their own eyes to provide both raw terrain information as well as the interpretation of that information for flight control. For optionally piloted or otherwise unmanned helicopters, there is a need to provide this same NOE capability with the inclusion of sensor(s) to gather terrain/ obstacle information, along with the appropriate guidance and control methods to make use of it.

Automatic flight of helicopters in the presence of obstacles has been explored by a number of researchers. As part of the DARPA Sandblaster program, Sikorsky Aircraft has flight demonstrated an integrated flight controls, sensor, and display system that is capable of automated approach to a point; but with some pilot intervention (Ref. [Sandblaster]). Vision-based methods are of interest because they are potentially light weight, inexpensive, and passive. Larger aircraft, on the other hand, due to their payload capability can use active sensors, such as LADAR or radar. Scherer et. al. [2] specifically used a custom 3D laser scanner to fly in an urban setting at speeds up to 10 m/s.

Under this effort, a number of sensor modalities have been considered, including radar, sonar, LIDAR/LADAR, and vision (monocular and stereo) techniques. Our subsequent work has focused on LADAR, due to available accuracy, range, and update rate of existing off-the-shelf sensors to support flight test evaluation. To support evaluation of methods in

Presented at The AHS International Specialists' Meeting on Unmanned Rotorcraft,

simulation for trade studies and to prepare for flight testing, a detailed simulation model was developed for scanning LADAR systems, allowing several existing off-the-shelf models to be tested in a closed loop simulation environment (Hokuyo UTM-30LX, Sick LMS291-S05, and Sick LD-MRS). Based on factors such as maximum range, weight, power, and field of view: the Sick LD-MRS system was then selected for further development and flight test validation of an automatic NOE flight system on a small, unmanned helicopter.

The remainder of this paper is organized as follows. First, two of the methods for generating the desired path to avoid obstacles are described. Second, a description of the aircraft utilized for simulation and flight test evaluation is included. Third, simulation and flight test results are discussed.

Guidance and Path Generation

Two primary methods for providing the guidance and path generation are explored here: (1) a simple processing of each laser scan and, (2) a potential-field method. The former is a relatively simple 2D method, working in the vertical plane. The later is computationally more expensive, and has been evaluated as both a 2D and 3D method.

Simple Processing of Single Scan Method: Here, the laser scanner is mounted such that a terrain profile is measured from directly below the helicopter to out in front of the helicopter, normal along the direction of travel, as far above the horizon as possible. In the case of the Sick LD-MRS, this translates to a field of view encompassing the bottom of the helicopter up to approximately 20 degrees above the horizon. In the simple processing method, each data set from the laser is first converted to a set of 3D points in an Earth fixed frame. The projected horizontal flight path is then compared to every point in the most recent laser scan. To ensure that the future path does not collide with any identified obstacle, a height restriction is then applied to each known point. The combination of observed obstacle points and height restrictions defines potential future trajectories. Altitude and vertical speed commands are then modified to achieve obstacle avoidance. The method pre-supposes an altitude control law that can track a specified altitude and vertical speed command. Here, altitude is the primary variable tracked. The vertical speed command is used to provide an additional feedforward signal to the controller for improved altitude tracking.

For scan points out in front of the aircraft, a minimum height restriction based on scan point *i* is found by:

$$h_{\min_i} = h_i + \Delta h_{desired} - \tfrac{1}{2} a_{desired} \Delta t_i^2$$

where $\Delta h_{desired}$ and $a_{desired}$ are specified vertical miss distance and desired maneuver vertical acceleration respectively, and Δt_i is the time remaining until the aircraft will be within the specified horizontal miss distance of scan point *i*. The commanded altitude is enforced as the maximum of the current command and the minimum from all scan points. A similar action occurs for vertical speed command as well:

$$\dot{h}_{\min_i} = -\sqrt{4a_{desired}\left(h - h_i - \Delta h_{desired}\right) + 2\left(a_{desired}\Delta t_i\right)^2} + a_{desired}\Delta t_i$$

where *h* is the current altitude of the helicopter. This expression ensures both a smooth pull up at the desired maneuver acceleration and a push-over at the top with the same acceleration. Or, if the current altitude is low enough that the aircraft cannot smoothly pull up at the specified vertical acceleration level ($h < \min_i h_{\min_i}$), then this same limit is found instead by:

$$\dot{h}_{\min} = -a_{desired}\Delta t_i$$

For scan points within a specified horizontal miss distance of the aircraft (i.e., points below the aircraft) these same formulae are used, but the time remaining is calculated based on capturing the desired minimum altitude using the specified vertical acceleration:

$$\Delta t_i = \begin{cases} \sqrt{\left|\dfrac{2\left(h - h_i - \Delta h_{desired}\right)}{a_{desired}}\right|}, & h > h_i + \Delta h_{desired} \\ 0, & otherwise \end{cases}$$

When the range of the terrain sensor is sufficient for the given terrain profile and selected vertical acceleration levels, this simple method provides commanded altitude and vertical speed to meet prescribed miss distances and vertical acceleration levels.

As described, this method can be utilized to modify any guidance policy to ensure the vertical profile does not come within specified distance of terrain. That is, act as a ground collision avoidance system. For true NOE flight, the nominal profile is set to be a specified nominal vertical descent rate. This combination of a nominal descent rate and ground collision avoidance logic results in NOE flight; at least in the vertical plane.

An important limitation of this method as evaluated is that it does not modify the horizontal speed of the aircraft, as would be necessary if a sufficiently tall obstacle appeared in the path. In principle, this method could be extended to modify the horizontal speed and heading of the aircraft as well. However, these extensions are not presented here.

Potential-Field Method: This method was developed as an appealing method to handle complex terrain/obstacle fields in a computationally efficient manner. Both 2D (vertical plane) and 3D versions have been developed. Here, each obstacle is considered a source while the end goal is considered a sink [3]. The aircraft then reacts to pseudo-forces acting on it by the sources and sinks. The method presupposes a control system that can track a desired position, velocity, and acceleration profile.

This task is accomplished by defining a map of the surrounding terrain features. The airspace around the aircraft is discretized and mapped to the array. Each element of the array is binary: 1 representing an occupied space, and 0 representing an empty space. Figure 1 shows a 2D example of this type of grid, called an occupancy grid. Each set of sensor measurements are used to update this array. This approach has the advantage that the size of the obstacle map is independent of sensor type and the number of sensor readings accumulated. Also, redundant sensor readings are easily included. This is a simplified version of the evidence grid technique [2].

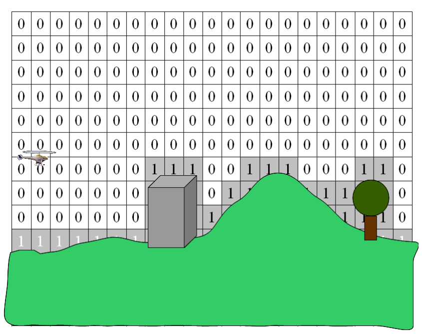

Figure 1. Obstacle map grid is utilized to register sensor returns (2D or 3D)

The path planner finds a smooth, continuous, obstacle-free path from the aircraft's current location to a desired waypoint. The mathematical machinery of potential theory provides a means to this end. In particular, the velocity field of an inviscid fluid flow around a body in the study of aerodynamics holds these characteristics, Such a situation can be represented as the gradient of a scalar potential function, φ:

$$V = \nabla\varphi\ .$$

Generally speaking, artificial potential field techniques formulate this problem by representing the goal point and obstacles as known spatial boundary conditions. The goal point Dirchlet condition on the potential function is set at -1 and the obstacles and space boundaries at 0.

The continuity equation, $\nabla \cdot V = 0$, reduces to Laplace's equation:

$$\nabla^2\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2} = 0$$

A finite difference approximation is applied to Laplace's equation to form a discrete potential field algebraic equation:

$$\frac{\partial^2\varphi}{\partial x^2} \approx \frac{\varphi_{i+1,j,k} - 2\varphi_{i,j,k} + \varphi_{i-1,j,k}}{\Delta x^2}$$

By making similar approximations in the y and z directions, and assuming an evenly spaced grid ($\Delta x = \Delta y = \Delta z$), and solving for $\varphi_{i,j,k}$, one obtains:

$$\varphi_{i,j,k} = \frac{\varphi_{i+1,j,k} + \varphi_{i-1,j,k} + \varphi_{i,j+1,k} + \varphi_{i,j-1,k} + \varphi_{i,j,k+1} + \varphi_{i,j,k-1}}{6}$$

In other words, the value of each point in the discrete potential field is equal to the average of the six points around it.

Once the array specifying φ is found, the same finite difference approximation can be used to calculate the gradient vector at each discrete point in the field. The streamline is then calculated from the vehicle starting position by 4th order Runge-Kutta integration of the gradient vector field using linear interpolation. The trajectory follows the gradient to the point of lowest potential, the goal. Note that this algorithm only produces a path in space and does not address the speed at which to fly.

The solution to this boundary value problem requires a starting guess and an iterative process. As each cell is updated, its new value is in turn used to update subsequent cells. The number of iterations required to converge depends upon the size of the array, the quality of the starting guess, and the convergence criteria used to terminate the algorithm. The algorithm can be significantly sped-up by using techniques detailed by Scherer, et. al. [2], including multi-grid, iterating only until the solution has no local minima, and setting the starting guess as a previous solution to the obstacle field. An example two-dimensional potential field with two-dimensional obstacles is shown in Figure 2.

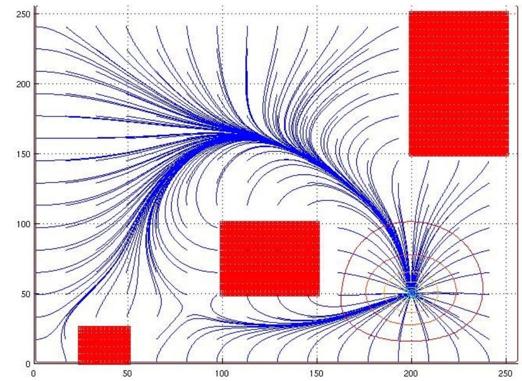

Figure 2. 2D potential field example with streamlines; obstacles are red boxes, goal in lower right corner

As stated above, the artificial potential field method provides only a path through the obstacle field but no details on what speed to use. Given a general twisting and turning obstacle free path, movement along this path at a constant velocity will cause changes in acceleration due to path curvature. In determining the speed to fly a particular path, maximum speed and acceleration limits are satisfied by the algorithm. These limits may be basic aircraft limits, limits fed back to the algorithm from the inner-loop flight controller, or limits imposed by an operator based on a given mission scenario. It may be desirable to traverse a commanded trajectory slowly for a given mission (for example, overwatch of a ground-based element) while very rapidly for another (for example, solo reconnaissance). Here, it is assumed that the dynamic constraints, such as maximum flapping angle, power output, etc. can be mapped to a maximum velocity and a maximum acceleration of the vehicle. These values are known prior to start of flight or fed to the algorithm by the underlying flight controller.

Given the geometric path, the speed shaping algorithm seeks to find a speed profile that traverses the path in the shortest time without violating dynamic constraints. The first step is to parameterize each coordinate by pathlength, *s*, with

$$\Delta s_i = \| r_i - r_{i-1} \|$$

$$s_i = s_{i-1} + \Delta s_i$$

where $s_0 = 0$. The unit tangent vector, unit normal vector, and the curvature are found as a function of *s* using finite difference approximations:

$$\vec{t}_i = \frac{\dfrac{\vec{r}_{i+1} - \vec{r}_{i-1}}{2 \cdot \Delta s_i}}{\left\| \dfrac{\vec{r}_{i+1} - \vec{r}_{i-1}}{2 \cdot \Delta s_i} \right\|}$$

$$\vec{n}_i = \rho \left(\frac{t_{i+1} - t_{i-1}}{2 \cdot \Delta s_i} \right)$$

$$\rho = \frac{1}{\left\| \dfrac{\vec{t}_{i+1} - \vec{t}_{i-1}}{2 \cdot \Delta s_i} \right\|}$$

To compute the speed profile, an initial guess for speed is found, typically just greater than zero to ensure that the initial guess does not violate any dynamic constraint. Using that initial guess, time is found as a function of pathlength.

$$\Delta s_i = \frac{v_i + v_{i-1}}{2} \cdot \Delta t_i$$

$$\Delta t_i = 2 \cdot \left(\frac{1}{v_i + v_{i-1}} \right) \cdot \Delta s_i$$

$$t_i = \sum_{j=1}^{i} \Delta t_j$$

Finally, the non-gravitational acceleration over the curve is found and added to gravity:

$$\vec{a}_i = \frac{\Delta v_i}{\Delta t_i} \cdot \vec{t}_i + \frac{v_i^2}{\rho_i} \cdot \vec{n}_i + \vec{g}$$

The guess values for $v(s)$ are then iterated point by point using the following logic: Does the point exceed specified constraint conditions? If yes, reduce the velocity at that point. Does the point exceed the overall velocity limit? If yes, reduce the velocity at that point. Does $\|a_i\|$ or $\|a_{i-1}\|$ exceed acceleration limits? If yes, reduce the velocity at that point. If the answers to the previous questions are no, then increase the velocity of the point. Once all the points on the velocity profile have converged, then every point in the velocity profile has met a constraint and the optimum has been found. Figure 3 shows an example solution.

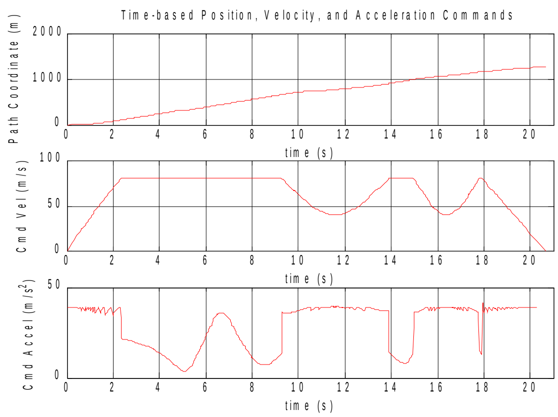

Figure 3. Speed shaping example: aircraft spends considerable time at maximum speed, but must at time decelerate to keep acceleration required to turn within limits

An operator supervising the aircraft requires a straightforward way to balance the competing desires to fly fast and to fly low based upon mission requirements. A straightforward approach in the potential field method context is to impose a virtual obstacle in the form of an artificial ceiling and floor into the potential field. With the artificial ceiling or floor, it is possible to cut off all paths from start point to goal. The solution to this problem is to create a "blanket" region around obstacles which overrides the imposed ceiling or floor. The aircraft is always left with a path over any obstacle if no lateral path exists.

A ceiling height is selected via the relationship:

$$h_C = h_{FA} - \kappa\left(h_{FA} - h_G\right)$$

where h_C is the ceiling height, h_G is the average height of the ground, and κ is masking factor. κ is set by the operator with a value between 0 to 1, with 0 being no masking and 1 being maximum masking. Note that because of the boundary conditions in the path planning algorithm, there is always a ceiling at the top of the flight area, here denoted at h_{FA}.

The blanket area is calculated by starting with the occupancy grid and propagating the occupancy grid one grid square/cube at a time until sufficient clearance has been achieved. The blanket must extend out at least twice the desired standoff distance from an obstacle, since the streamline will be halfway between the ceiling and the obstacle. Finally, the blanketed volume is subtracted from the ceiling to produce a modified ceiling. This ceiling is added to the obstacle map and is otherwise treated as an obstacle.

Note that masking is inversely related to speed; a masked path will tend to produce sharper bends in the planned path, and the velocity planner accordingly slows the aircraft to accomplish such turns.

Test Aircraft

A Yamaha RMAX based research UAV, Figure 4, was utilized for the simulation and flight test activities under this effort. The system consists of four major elements: the basic Yamaha RMAX airframe, a modular avionics system, baseline software, and a set of simulation tools.

Figure 4. Yamaha RMAX based research UAV utilized for this effort, 10.2 ft main rotor

The hardware components that make up the baseline flight avionics include general purpose processing capabilities and sensing. The research avionics configuration includes:

- 2 Embedded PCs
- Inertial Sciences ISIS-IMU Inertial Measurement Unit
- NovAtel OEM-4, differential GPS
- Sick LD-MRS laser scanner, Figure 5
- Custom made ultra-sonic sonar altimeter
- Honeywell HMR-2300, 3-Axis magnetometer
- Actuator control interface
- Vehicle telemetry (RPM, Voltage, Remote Pilot Inputs, low fuel warning)
- 11 Mbps Ethernet data link and an Ethernet switch
- FreeWave 900MHz serial data link

Figure 5. Sick LD-MRS Laser scanner mounted under the nose of the aircraft, able to see down and forward (sensor rotated 90 degrees in roll, 40 degrees nose down pitch)

The baseline navigation system running on the primary flight computer is a 17 state extended Kalman filter. The states include: vehicle position, velocity, attitude (quaternion), accelerometer biases, gyro biases, and terrain height error. The system is all-attitude capable and updates at 100 Hz [4]. The baseline flight controller is an adaptive neural network trajectory following controller with 18 neural network inputs, 5 hidden layer neurons, and 7 outputs for each of the 7 degrees of freedom [5]. These 7 degrees of freedom include the usual 6 rigid-body degrees of freedom plus a degree of freedom for rotor RPM. The baseline flight controller and navigation system, which coupled with the simple baseline trajectory generator, is capable of automatic takeoff, landing, hover, forward flight up to the maximum attainable by the helicopter (around 85 feet/sec) and aggressive maneuvering.

Simulation Results

Flight control software was developed utilizing the existing Georgia Tech UAV Simulation Tool (GUST), which is a set of C/C++ software that supports pure software, hardware-in-the-loop, and research flight test operations [6]. GUST includes models of the sensors, aircraft, and aircraft interfaces – down to the level of binary serial data (i.e., packets). It enables injection of model error and environmental disturbances. It includes a flexible scene generation capability and reconfigurable data communication routines, enabling a large number of possible hardware-in-the-loop simulation configurations. Under this effort, a detailed sensor model for the Sick LD-MRS was added to this environment.

Simple Processing of Single Scan Method: The single scan method was tested for a closed course at variety of speeds (20-50 ft/sec) and desired altitudes above obstacles (20 to 50 feet). For sake of comparison, only the 40 ft/sec / 50 foot case is shown here; as these correspond to the flight test data also available.

Figure 6 shows a 3D plot of the recorded trajectory for two passes around the simulated closed course. Note that the single obstacle in the path, representing a tree line at the flight test location, results in significant flight path alternation, and the specified horizontal and vertical miss distances are satisfied (50 feet each). Figure 7 shows the altitude above a reference height (corresponding to approximately the terrain height for most of the field) and vertical speed vs. time profile for one of the passes over the simulated tree line (modeled as a box with appropriate length, width, height, and location).

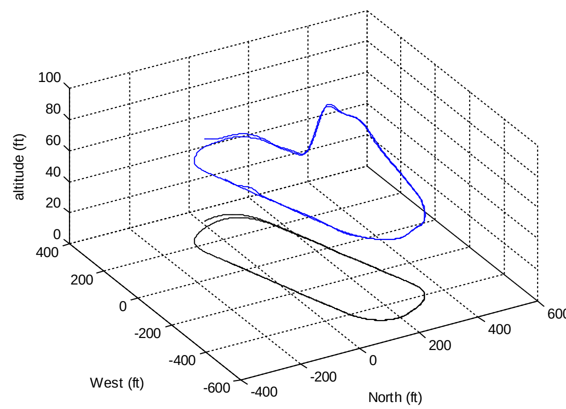

Figure 6. Simulation results with single scan algorithm 40 ft/sec, desired terrain height of 50 feet, for two rounds of a closed course (horizontal projection of path shown on bottom); dominant feature is the avoidance of simulated tree line as the aircraft traverses clockwise in the plot

Potential Field Method: The potential field algorithm was also tested in the full nonlinear simulation prior to flight test. The intent of simulation was to identify the path planner's performance in a practical environment, as well as to test the limits of the algorithm with regards to sensor performance. Table 1 lists the base parameters used. Note that the aircraft was commanded to yaw slowly from side to side to enable the aircraft to generate a 3D map.

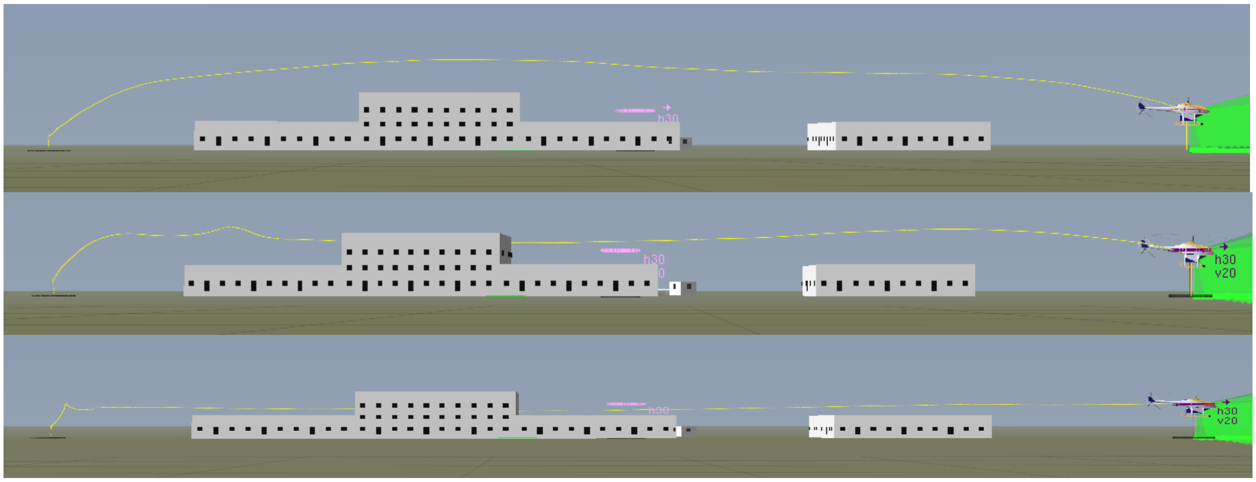

Figure 8. Simulation results with three different masking factors (0 top, 0.4 middle, 0.7 bottom); aircraft not drawn to scale, tallest building 50 ft high

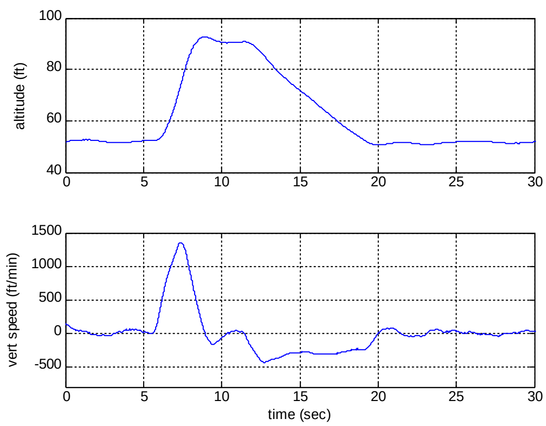

Figure 7. Simulation results with single scan algorithm 40 ft/sec, desired terrain height of 50 feet, close up of pass of simulated tree line, desired descent rate after encounter was 300 ft/min

Table 1: Parameters for potential field simulations

Parameter	Value
Horizontal Grid Size	64
Vertical Grid Size	32
Horizontal Grid Resolution	15 feet
Vertical Grid Resolution	5 feet
Sensor Range	200 feet
Sensor Field of View	-90 to 10 degrees
Masking Factor	0 to 0.8
Desired Speed	20 to 50 feet/second

The masking factor was varied to examine the performance of the algorithm in different masking conditions. During flight, the potential of the space above an obstacle has either equal or lower potential than the space to the sides. If the aircraft were to start particularly close to the top of the obstacle, it would tend to fly over it. However, if the potential above the obstacle increased, as it does when the high masking factor/lower ceiling is imposed, the vehicle tends toward the sides of the obstacle. The same effect happens if the aircraft is simply presented with a taller obstacle. Simulation confirmed this expectation. Example results showing the contrast between runs at the same obstacle for masking factors of 0.0, 0.4, and 0.7 are shown in Figure 8. In the no masking case, the aircraft flies directly over the top of the obstacles. In the case of 0.4 masking, the aircraft initially climbs over the low part of the obstacle, but then laterally avoids the taller part. In the case of 0.7 masking, the aircraft takes a wide berth of all the obstacles, while maintaining a low, masked profile.

The mission speed was varied in another set of tests to see how the planner would react while travelling at progressively faster speeds. The expectation was that as the speed increased, the aircraft would get closer and closer to colliding with the obstacle. Eventually it was expected to detect obstacles without enough time to plan a new path and/or decelerate to a stop. Three example cases are shown: 30 ft/s, 40 ft/s, and 50 ft/s, Figure 9. All cases were run at 0.2 masking. In the first case, the planner has plenty of time to avoid the obstacle with little deceleration—primarily a change in direction. In the second case, the aircraft measures the obstacle, but needs to slow to a near stop before turning and assuming a new direction of travel. In the final case, the aircraft couldn't calculate a new path and accelerate into it before colliding with the obstacle.

Flight Test Results

Simple Processing of Single Scan Method: To date, the single scan method was tested for a closed course at variety of speeds (20-40 ft/sec) and a desired altitude of 50 feet. First over flat ground, then under conditions similar to those tested in simulation (encounter of tree line).

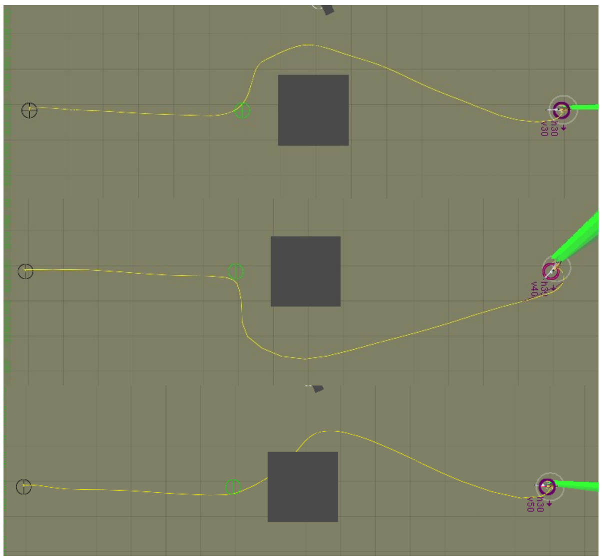

Figure 9. Simulation results with three mission speeds (30 feet/second top, 40 middle, 50 bottom); aircraft not drawn to scale, building 100 feet wide

Figure 10 shows a 3D plot of the recorded trajectory for two passes around the closed course, corresponding to Figure 6 for simulation results. The alternation in flight path is somewhat different, which is likely primarily due to differences in the actual geometry of the tree line. Specifically, the tree line in the simulated was modeled as a box, whereas the true tree-line was curved with only a single maximum altitude – presumably resulting in a somewhat more gentle avoidance maneuver. Figure 11 shows the altitude and vertical speed vs. time profile for one of the passes over the actual tree line, corresponding to Figure 7 from the simulator.

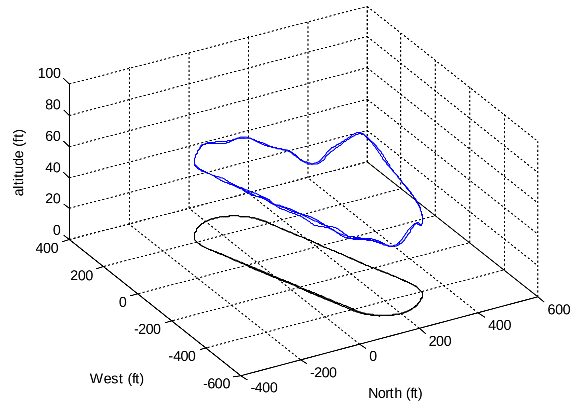

Figure 10. Flight test results with single scan algorithm 40 ft/sec, desired terrain height of 50 feet, for two rounds of a closed course (horizontal projection of path shown on bottom)

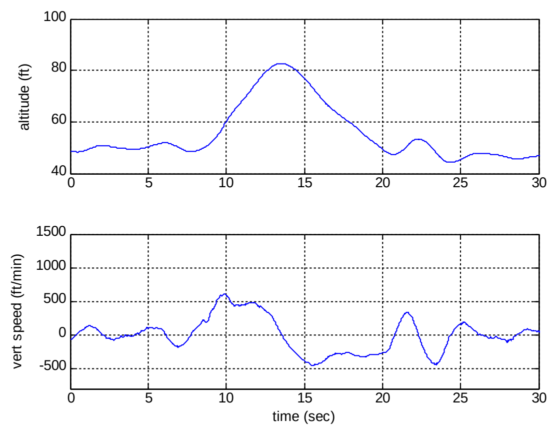

Figure 11. Flight test results with single scan algorithm 40 ft/sec, desired terrain height of 50 feet, close up of pass of simulated tree line; maneuver is more gentle than simulation, likely due to different geometry of the tree line (curved); also note secondary avoidance after 20 seconds, probably due to passing over a fence not in the simulation model

Potential Field Method: Flight testing to date has progressed as far as testing the algorithm against simulated obstacles using simulated sensor data in flight. These tests confirmed the basic capability of the potential field algorithm and partially validate the simulation results. Closed loop flights utilizing the actual sensor and actual obstacles are an area of future work.

Conclusions

The efforts described in this paper include: (1) Flight testing of installed ranging sensor, specifically the Sick LD-MRS; (2) Hardware-in-the-loop Simulation studies based on achieved sensor performance utilizing two methods for generating the desired NOE flight path; and (3) Flight testing of closed loop system performing autonomous unmanned NOE flight. Flight test results verify the effectiveness of the installed sensor, and validate the simulation results of the simpler algorithm up to 40 feet/second. Future work includes expanding the speed and acceleration levels.

Acknowledgements

The authors would like to acknowledge support from Sikorsky Aircraft Corporation and the Georgia Institute of Technology, and the help of other contributors including: JVR Prasad, Mark Costello, Jeong Hur, Brian McCabe, Russ Halstead, Suresh Kannan, Scott Kimbrell, and Wesley DeBusk.

References

[1] [sandblaster reference]

[2] S. Scherer, S. Singh, L. Chamberlain, M. Elgersma, "Flying Low and Fast Among Obstacles: Methodology and Experiments," The International Journal of Robotics Research, Vol. 27, No. 5, 549-574 (2008).

[3] C. W. Warren, "Global Path Planning Using Artificial Potential Fields," IEEE International Conference on Robotics and Automation, 1989.

[4] Dittrich, J.S. and Johnson, E.N., "Multi-Sensor Navigation System for an Autonomous Helicopter," Proceedings of the 21st Digital Avionics Systems Conference, October 2002.

[5] Johnson, E.N. and Kannan, S.K., "Adaptive Trajectory Control for Autonomous Helicopters," AIAA Journal of Guidance, Control, and Dynamics, Vol. 28, No. 3, pp. 524-538, May/June 2005.

[6] Johnson, E.N. and Schrage, D.P., "System Integration and Operation of a Research Unmanned Aerial Vehicle," AIAA Journal of Aerospace Computing, Information, and Communication, Vol. 1, No. 1, pp. 5-18, January 2004.